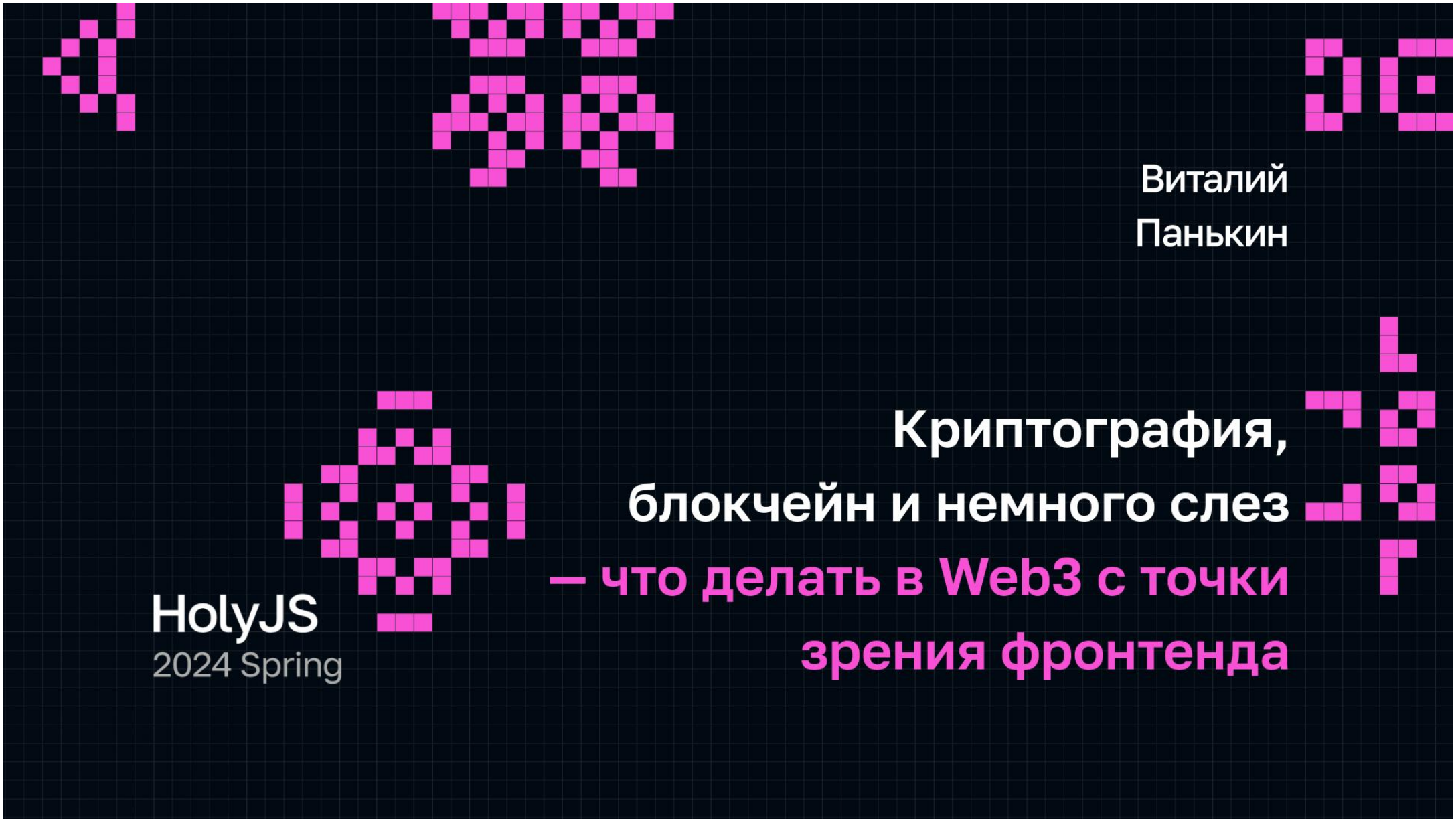
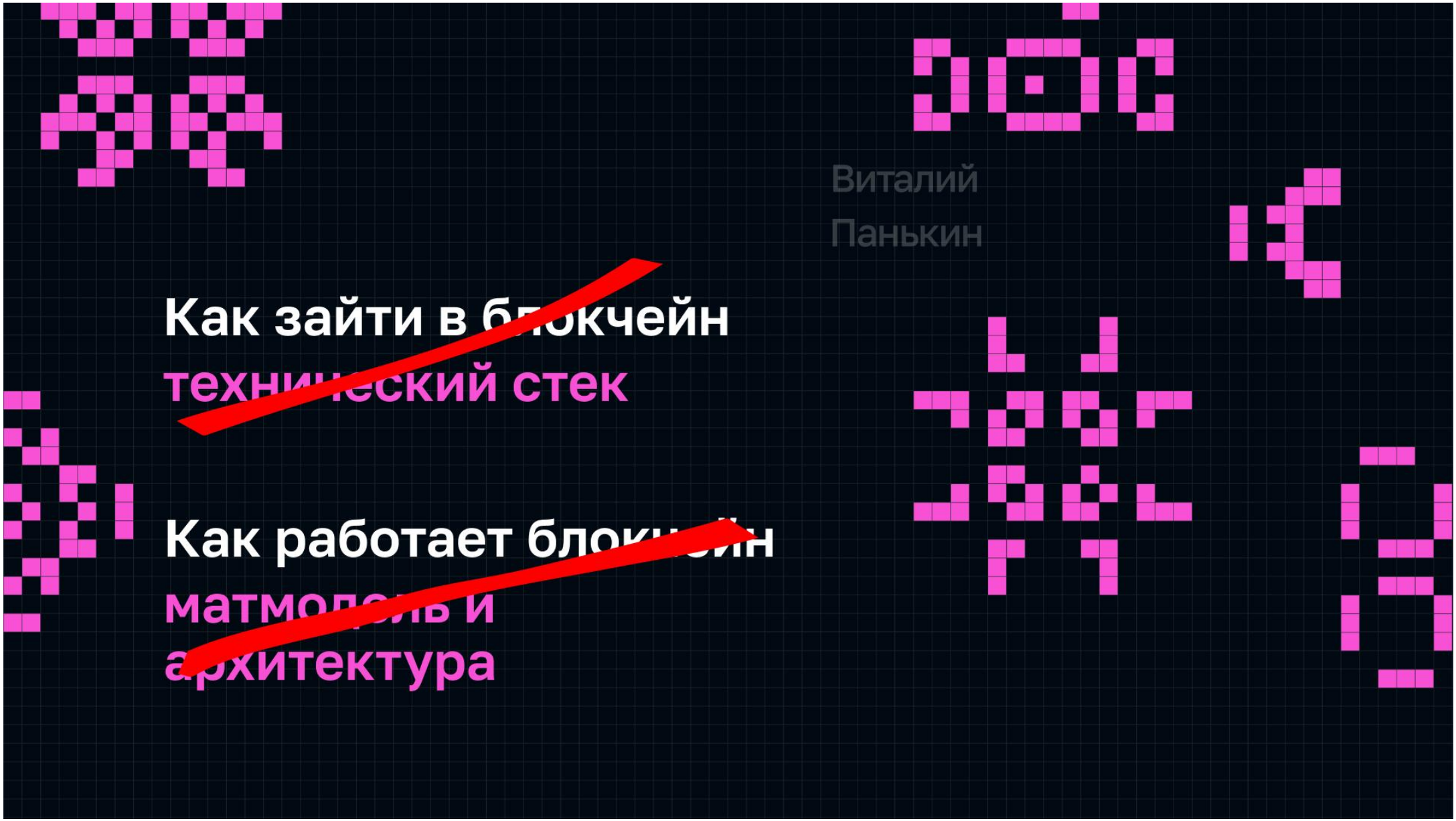
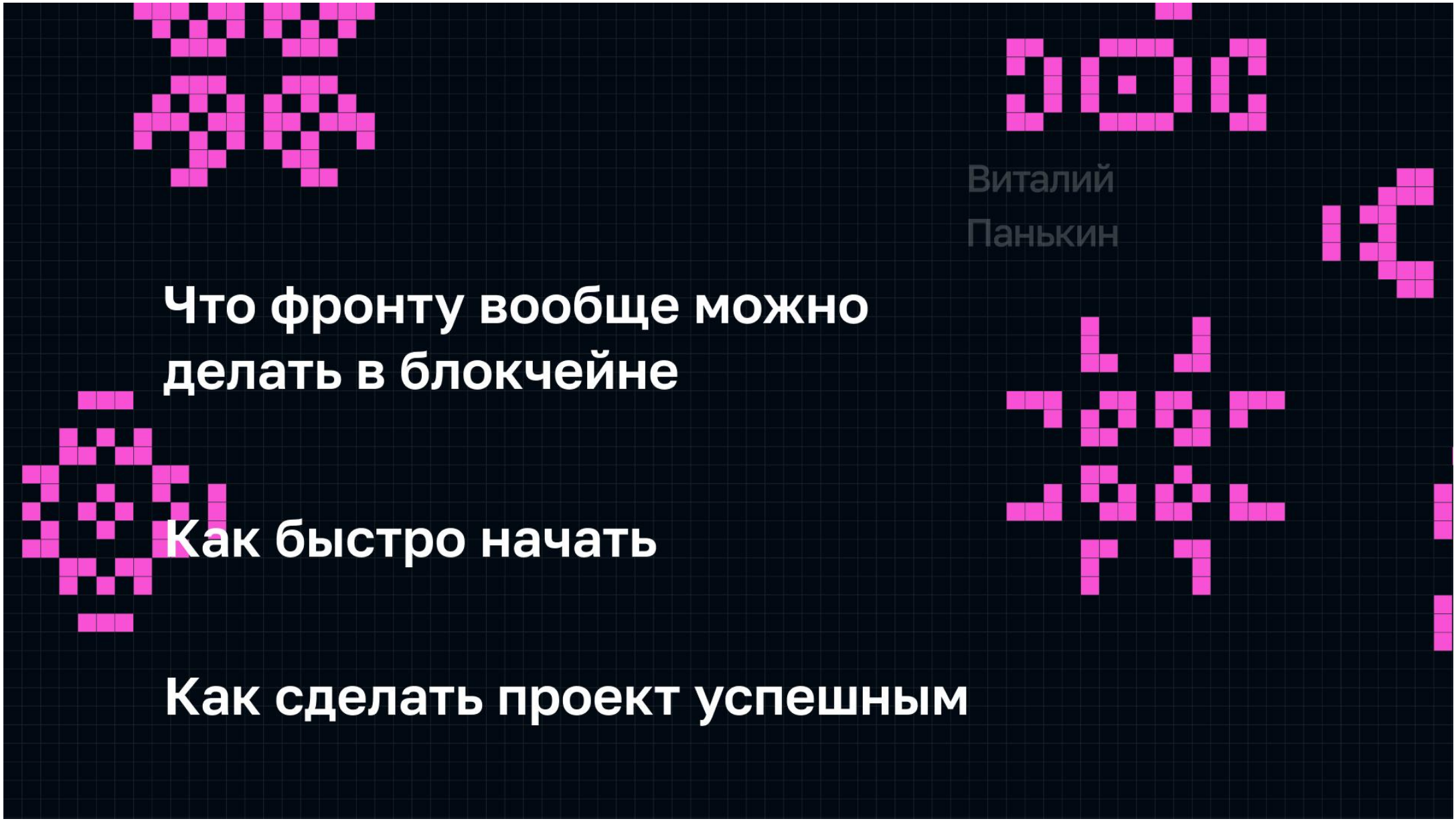

Виталий
Панькин
Криптография,
блокчейн и немного слез
— что делать в Web3 с точки
зрения фронтенда
HolyJS
2024 Spring


Виталий
Панькин
Как зайти в блокчейн
технический стек
Как работает блокчейн
математика и
архитектура


Виталий
Панькин
Что фронту вообще можно
делать в блокчейне
Как быстро начать
Как сделать проект успешным

Виталий
Панькин

Full stack developer

Работал с TVM сетями

Бился в RnD команде

Писал в Open Source

TON
Venom
Everscale

План

4 основных сценария

3 специальных случая

Бонусные рекомендации

Блокчейн

Пиринговая сеть

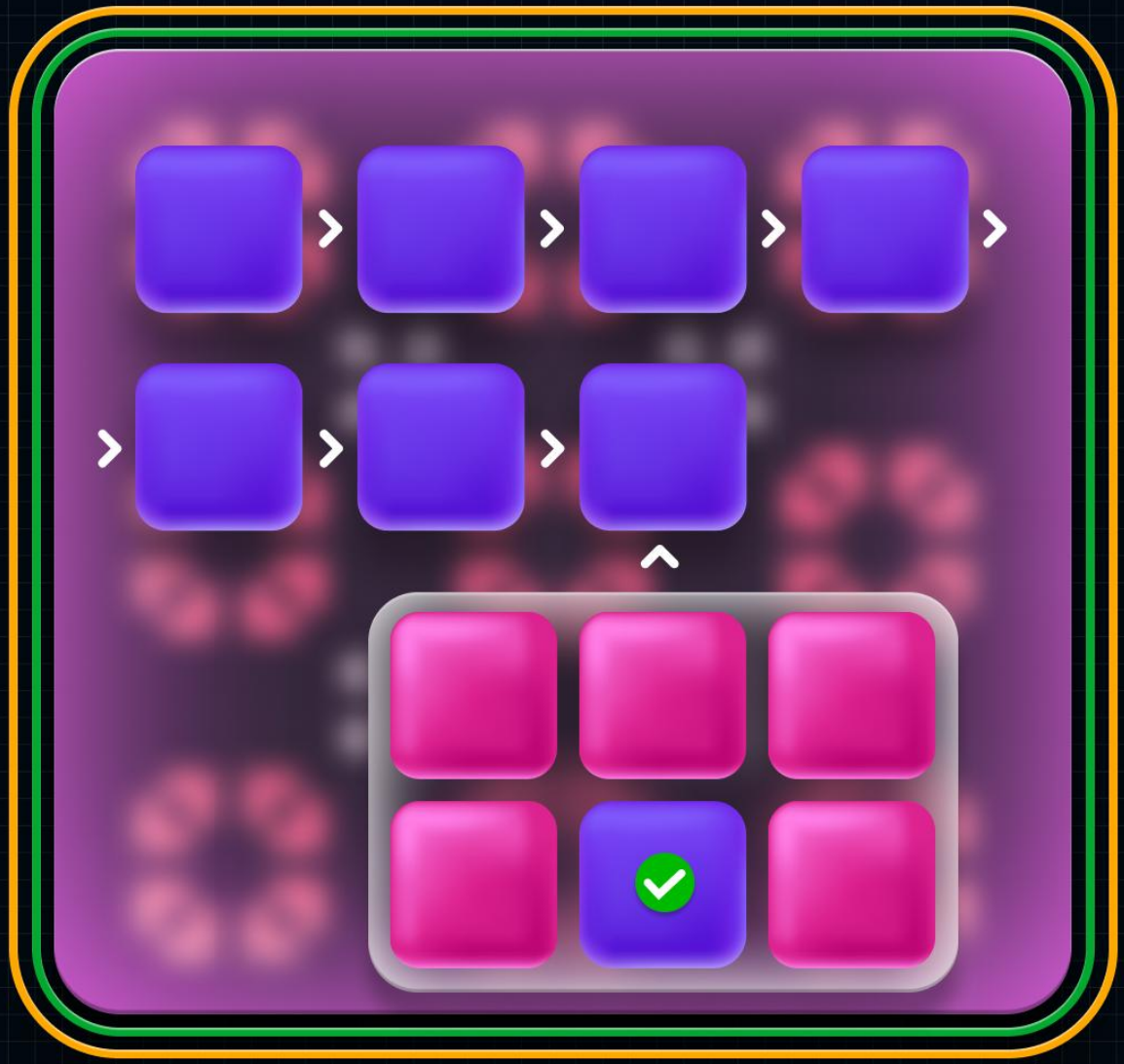
общается между собой

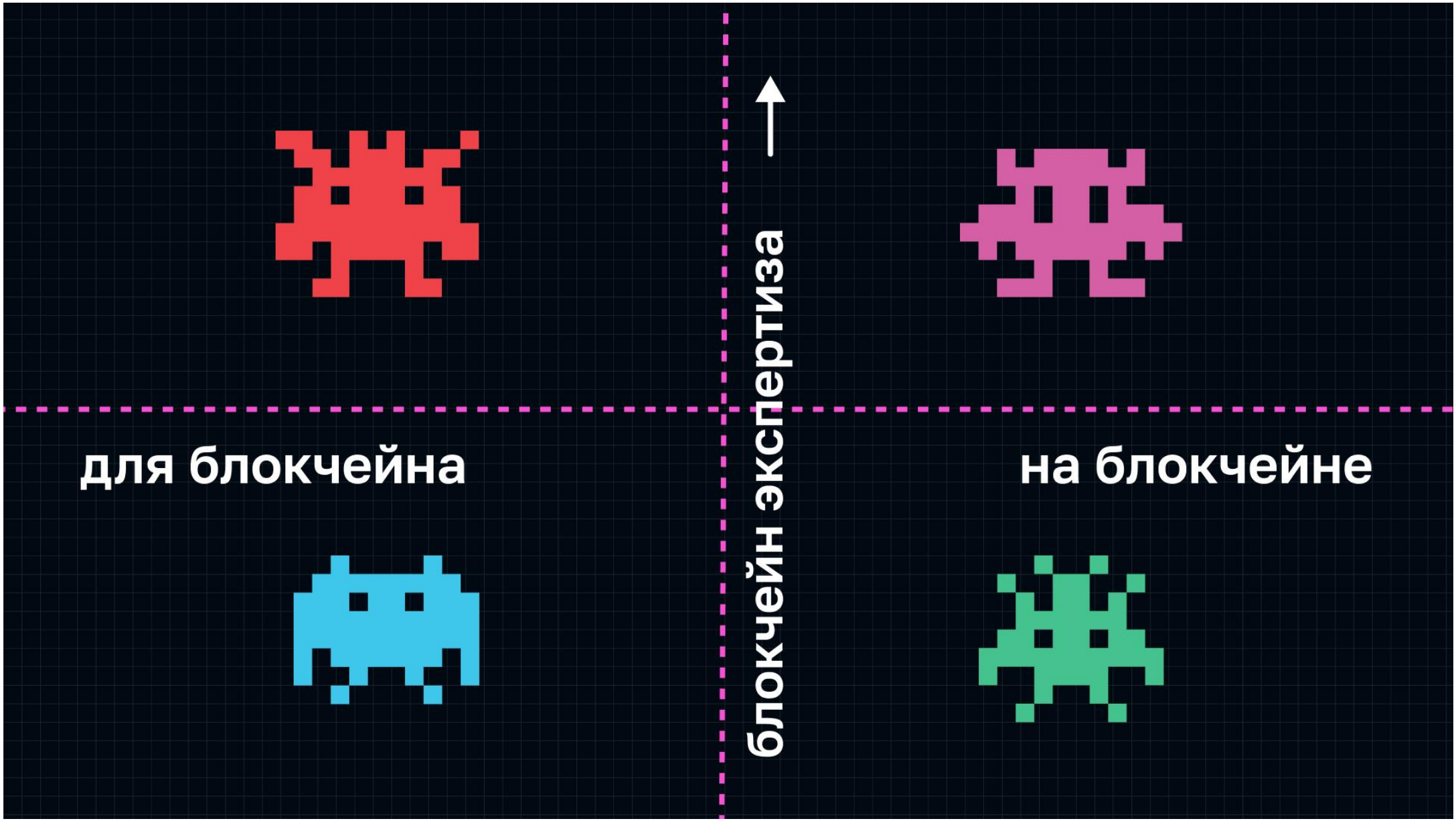
и по алгоритму консенсуса

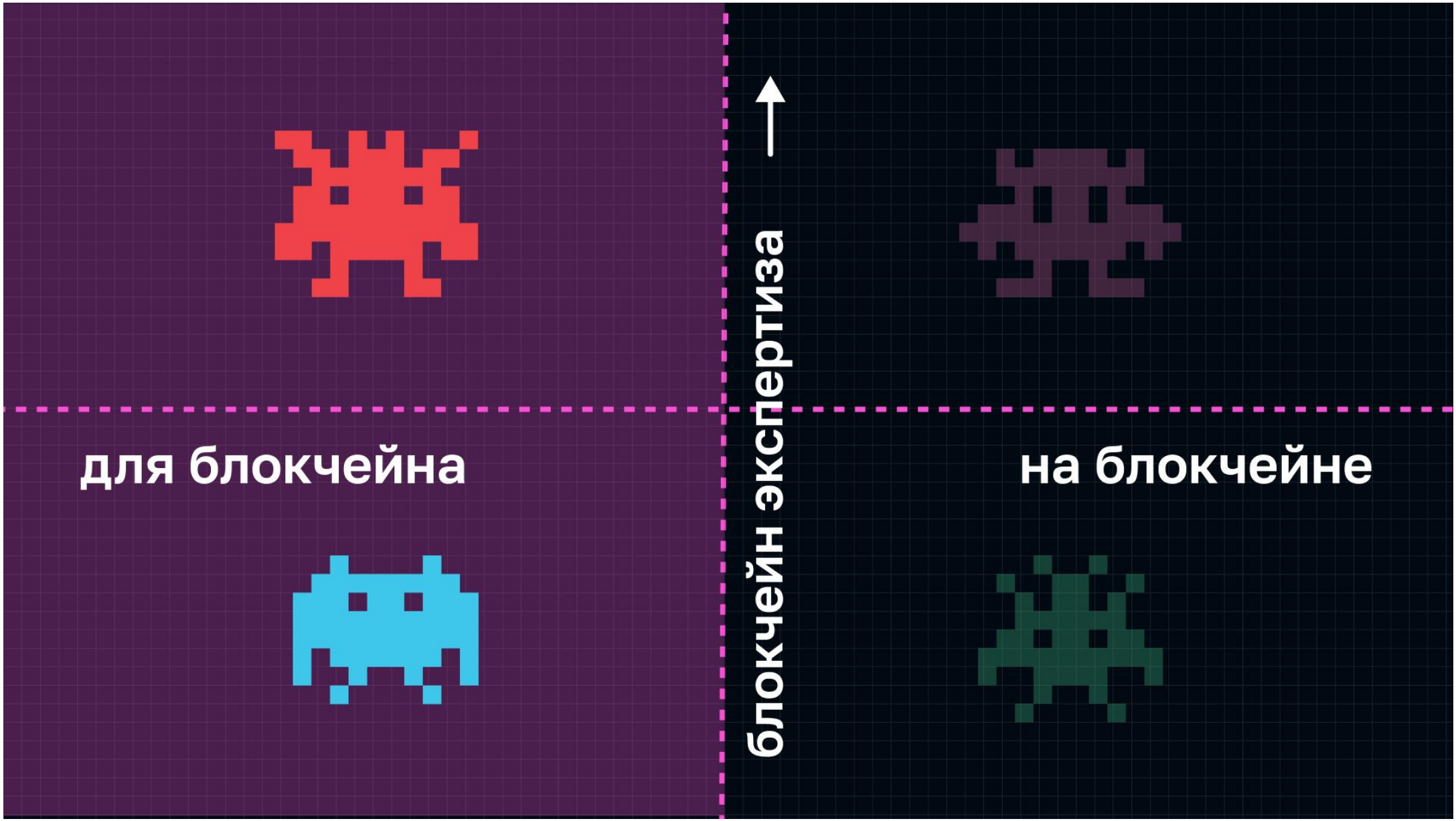
используя хэш + ЭЦП

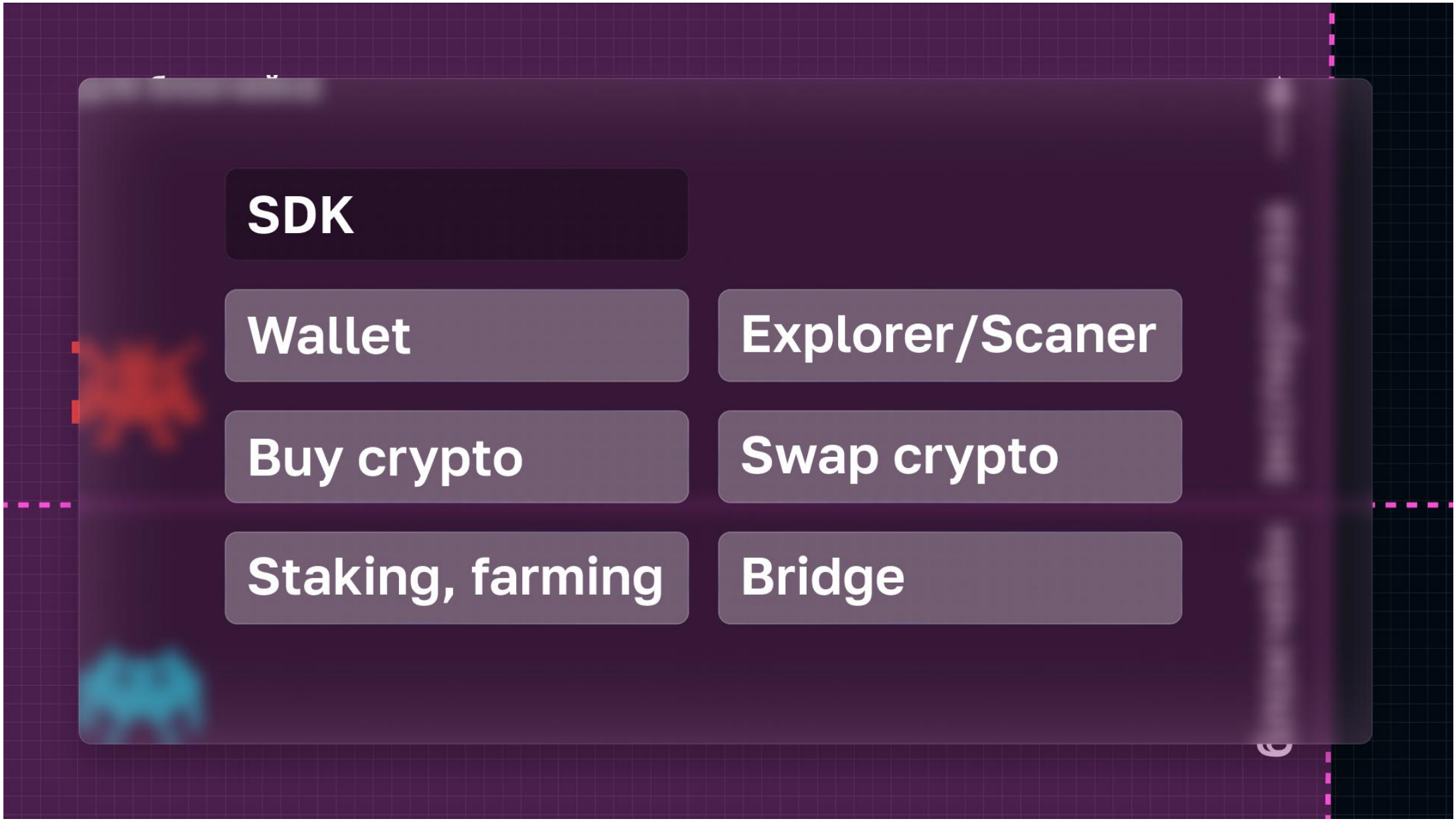
сохраняет данные в блоки

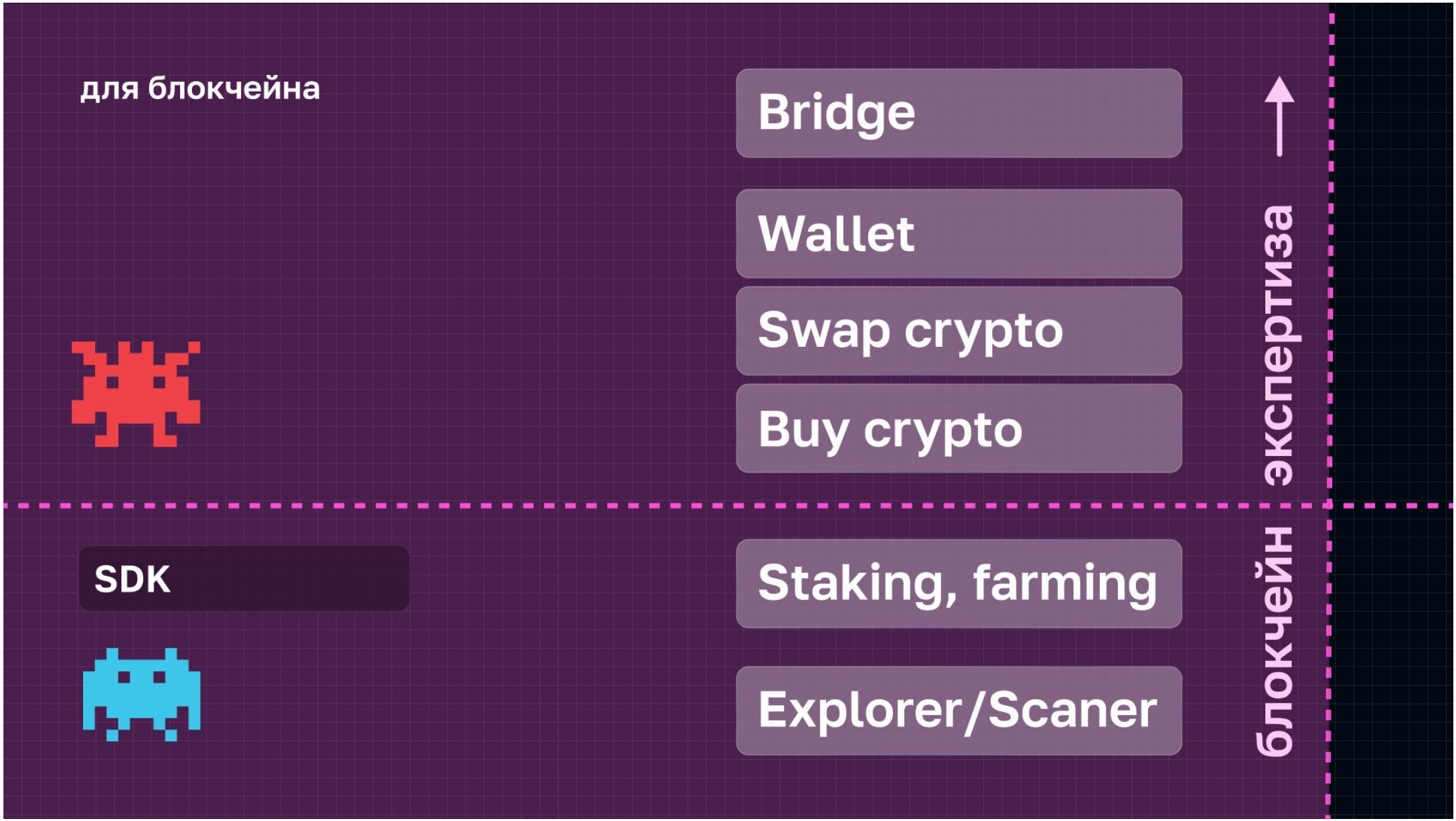
которые нельзя подделать




блокчейн экспертиза
для блокчейна
на блокчейне


блокчейн экспертиза
для блокчейна
на блокчейне


SDK
Wallet
Explorer/Scaner
Buy crypto
Swap crypto
Staking, farming
Bridge


для блокчейна
Bridge
Wallet
Swap crypto
Buy crypto
экспертиза
SDK
Staking, farming
Explorer/Scaner
блокчейн


ethereum
web3.js
ethers.js
BINANCE SMART CHAIN
web3.js
Polkadot
polkadot.js
SOLANA
@solana/web3.js
CARDANO
cardano-wallet-js

ethereum

web3.js

ethers.js

```
import Web3 from 'web3';

// Получаем баланс по адресу
async function getBalance(address) {
    try {
        // Подключаемся к Ethereum ноде
        const web3 = new
Web3('https://mainnet.infura.io/v3/YOUR_INFURA_PROJECT_ID');

        // Получаем баланс
        const balanceWei = await web3.eth.getBalance(address);

        // Конвертируем баланс из Wei в Ether
        const balanceEther = web3.utils.fromWei(balanceWei, 'ether');

        console.log(`Balance of ${address} is: ${balanceEther} ETH`);
    } catch (error) {
        console.error(`Error fetching balance: ${error}`);
    }
}

// Пример работы (тут должен быть реальный адрес)
const address = '0x0000000000000000000000000000000000000000';
getBalance(address);
```

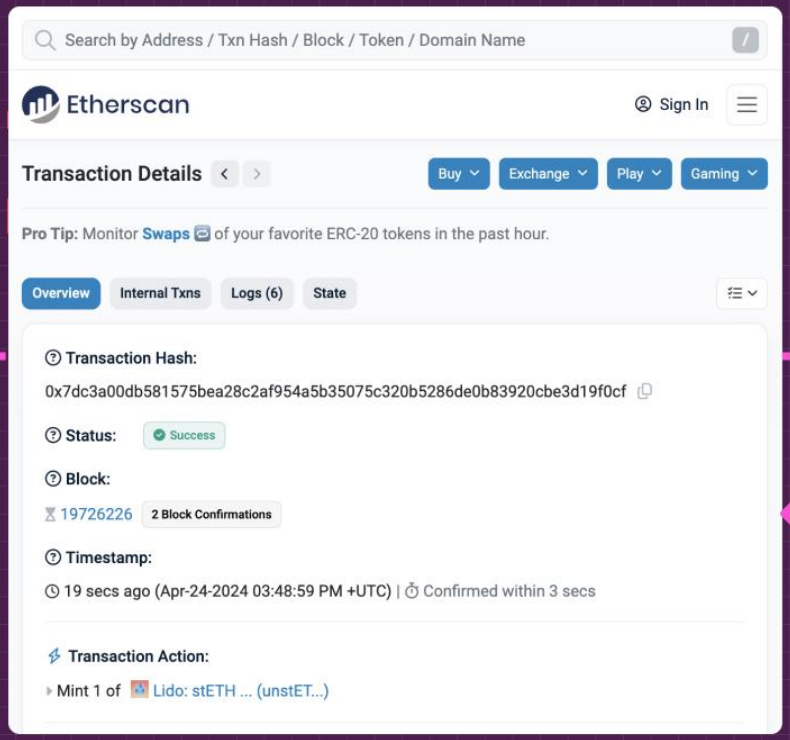
ethereum

web3.js

ethers.js

```
import ethers from 'ethers';

// Получаем баланс по адресу
async function getBalance(address) {
    try {
        // Подключаемся к Ethereum сети по умолчанию (mainnet)
        const provider = new ethers.providers.getDefaultProvider();

        // Получаем баланс
        const balance = await provider.getBalance(address);

        // Конвертируем баланс из Wei в Ethereum
        const balanceInEther = ethers.utils.formatEther(balance);

        console.log(`Balance of ${address} is: ${balanceInEther} ETH`);
    } catch (error) {
        console.error(`Error fetching balance: ${error}`);
    }
}

// Пример работы (тут должен быть реальный адрес)
const address = '0x0000000000000000000000000000000000000000';
getBalance(address);
```

для блокчейна



- Bridge
- Wallet
- Swap crypto
- Buy crypto
- Staking, farming
- Explorer/Scaner

экспертиза блокчейн


SOLIDITY
vyper


```
pragma solidity ^0.8.0;

contract Token {
    string public name = "Holy Token";
    string public symbol = "HOLY":
    uint256 public supply = 1000000;
    address public owner;
    mapping(address => uint256) balances;

    constructor() {
        // The supply is assigned to transaction sender,
        // which is the account that is deploying the contract.
        balances[msg.sender] = supply:
        owner = msg.sender;
    }
    function transfer(address to, uint256 amount) external {
        require (balances [msg.sender] >= amount, "Not enough tokens");
        balances[msg.sender] -= amount;
        balances[to] += amount;
    }
    function balanceOf(address account) external view returns (uint256) {
        return balances [account];
    }
}
```

похож на singleton

после деплоя лежит лежит как микробэк по какому-то адресу

имеет интерфейс взаимодействия abi.json

```sol
pragma solidity ^0.8.0;

contract Token {
    string public name = "Holy Token";
    string public symbol = "HOLY":
    uint256 public supply = 1000000;
    address public owner;
    mapping(address => uint256) balances;

    constructor() {
        // The supply is assigned to transaction sender,
        // which is the account that is deploying the contract.
        balances[msg.sender] = supply:
        owner = msg.sender;
    }
    function transfer(address to, uint256 amount) external {
        require (balances [msg.sender] >= amount, "Not enough tokens");
        balances[msg.sender] -= amount;
        balances[to] += amount;
    }
    function balanceOf(address account) external view returns (uint256) {
        return balances [account];
    }
}
```

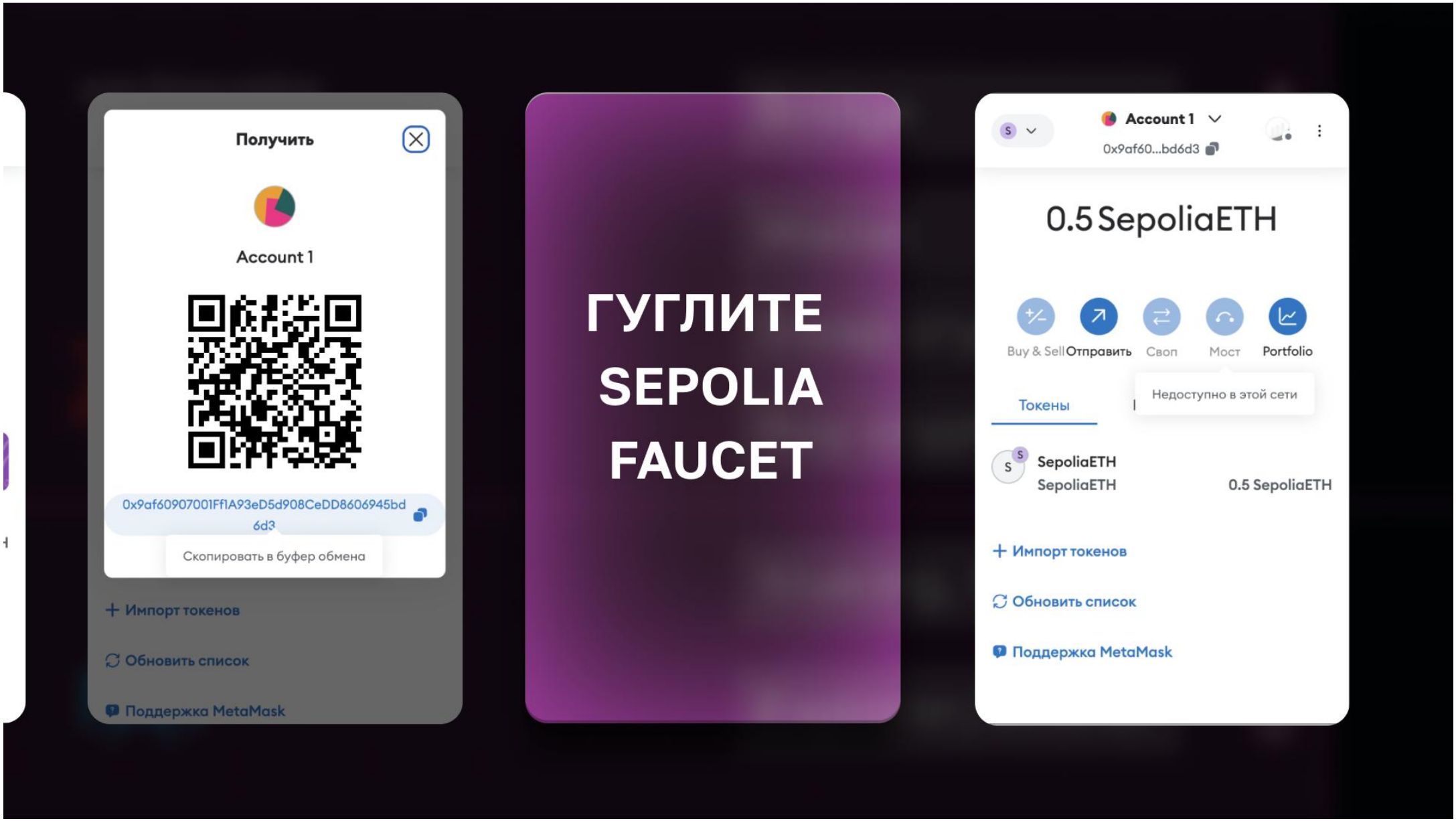
онлайн IDE

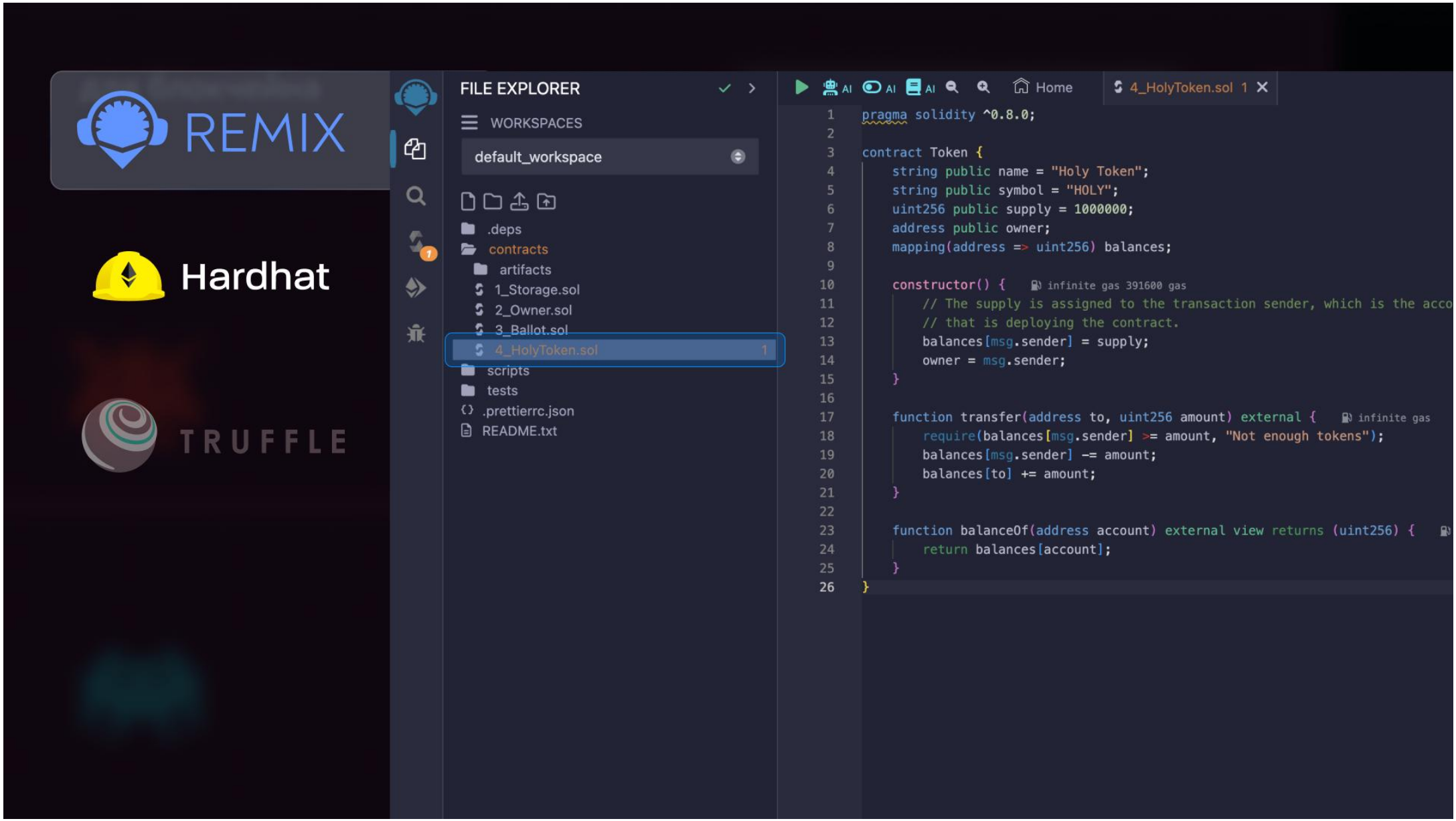
готов к использованию сразу
ниже порог входа
менее гибок, но лучше для обучения

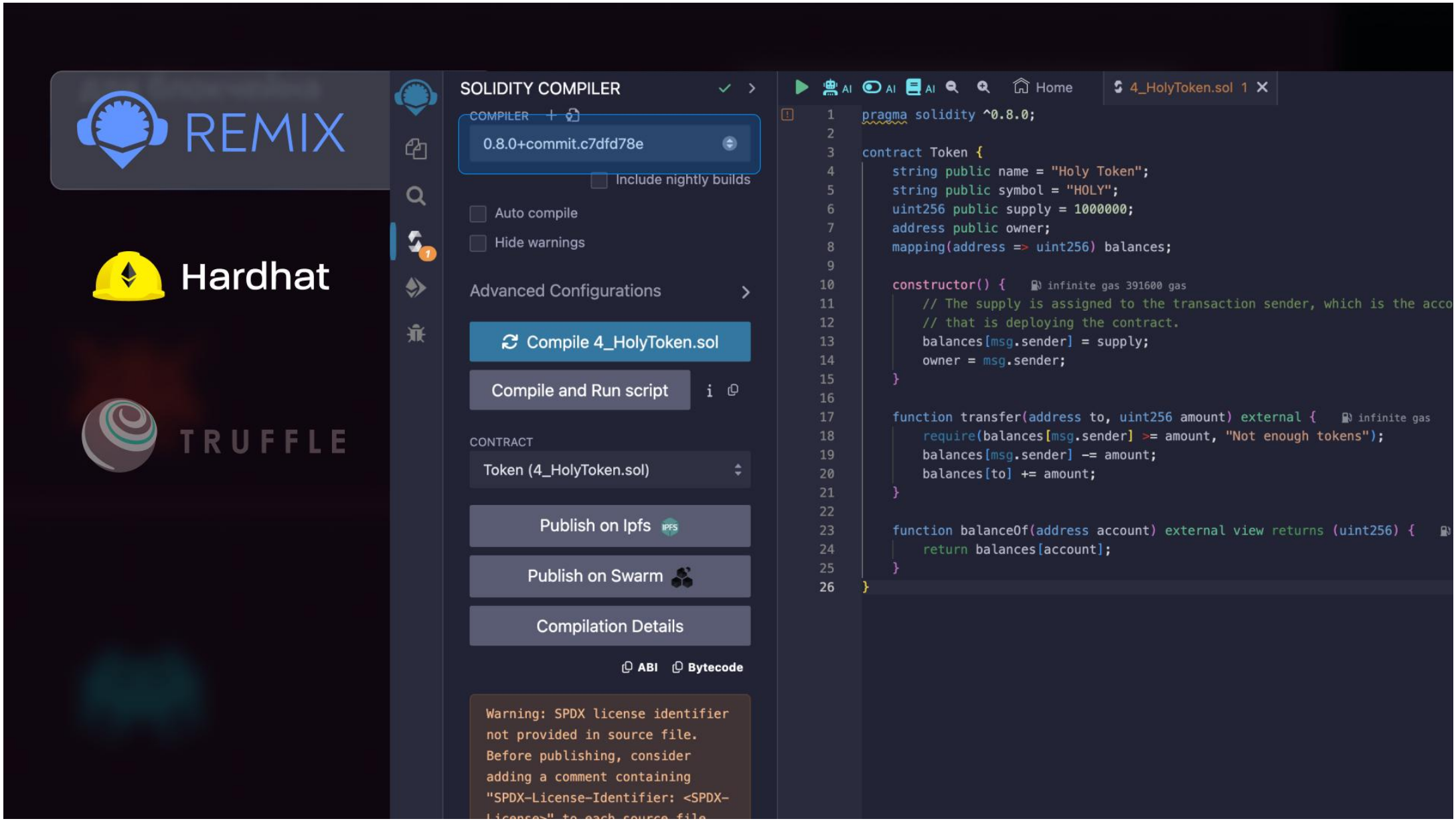
консоль и код

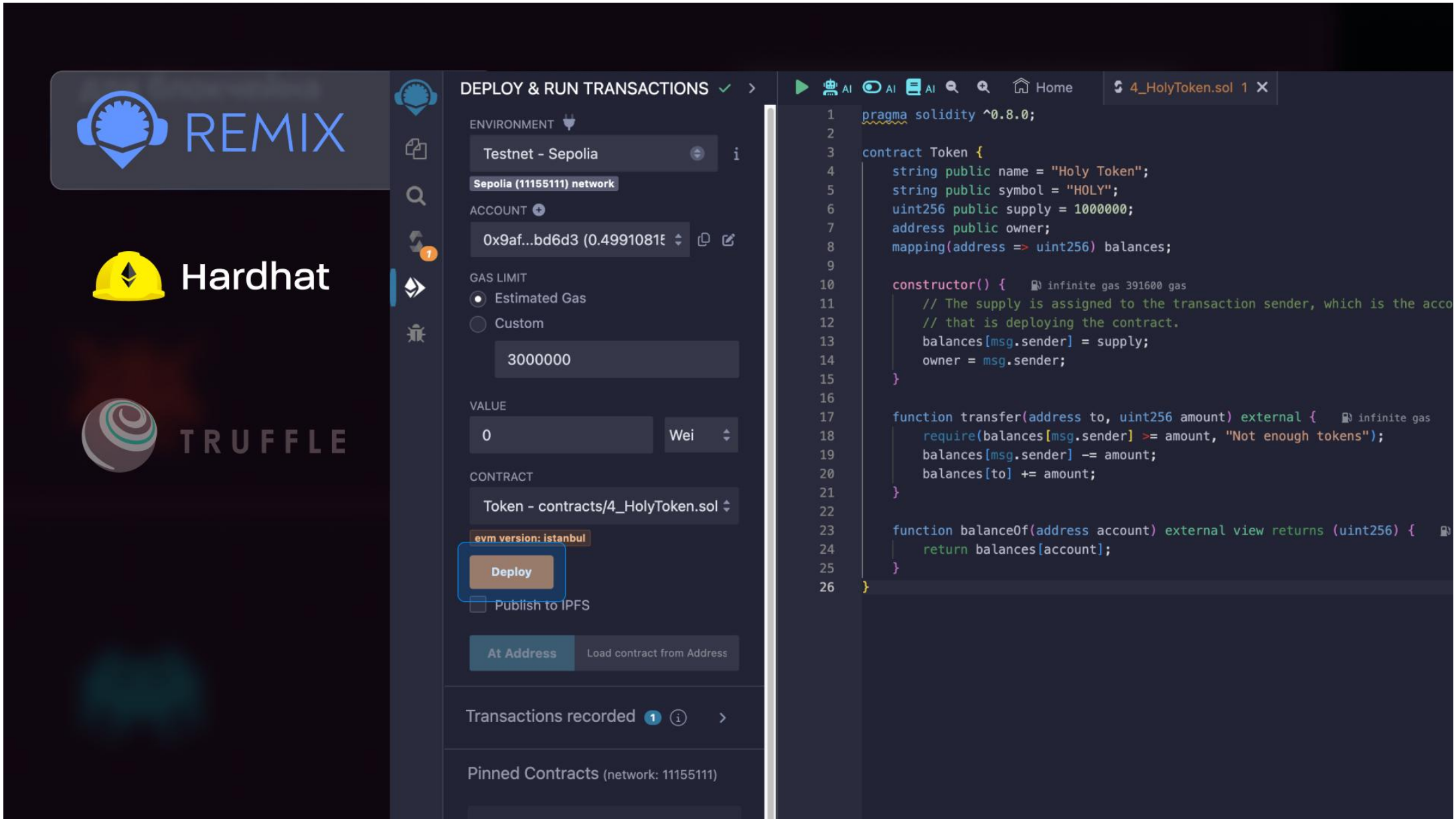
похож на стандартый модульный npm подпроект
есть понятный флоу тестирования
для старта чуть сложнее

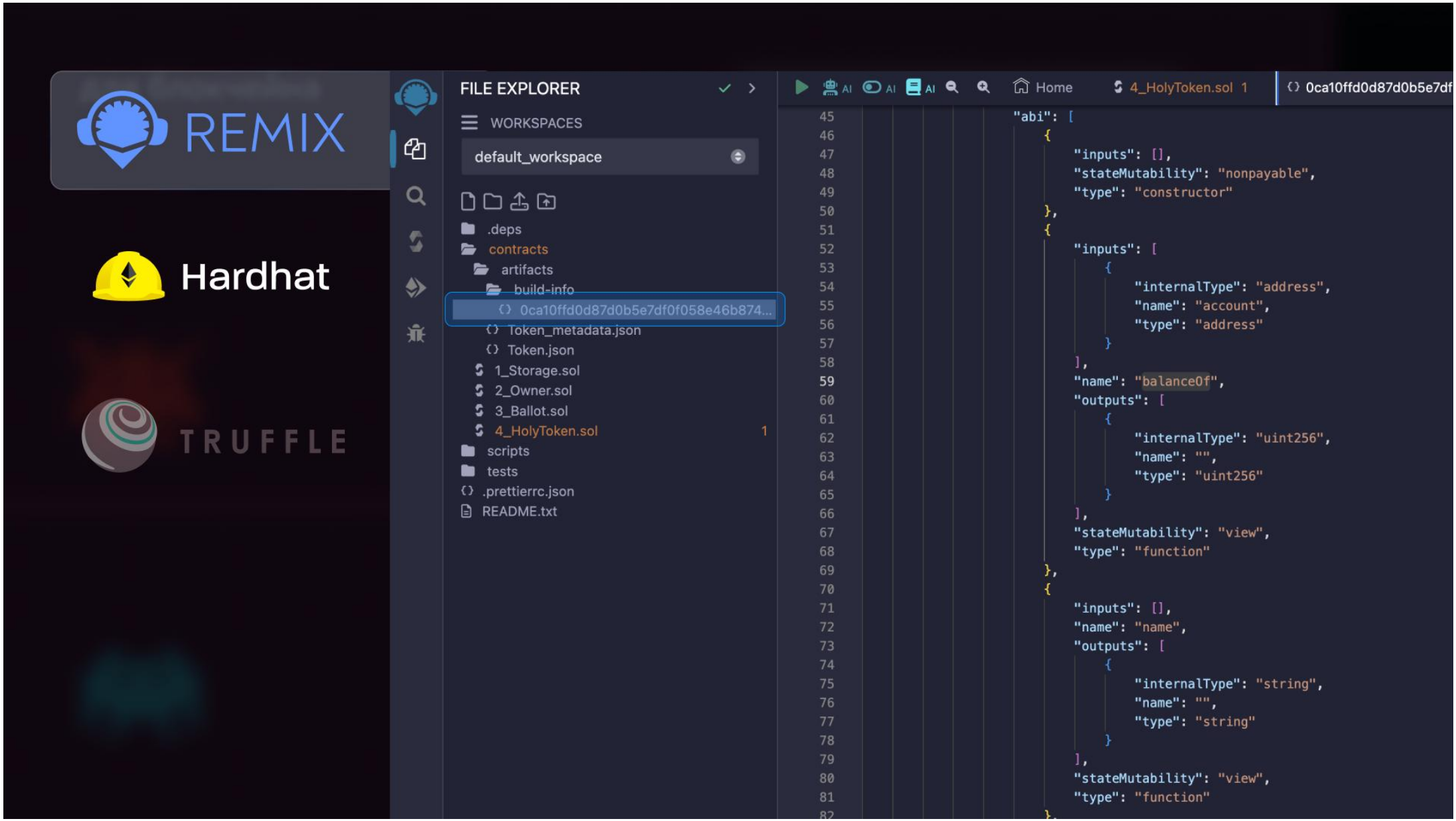
загрустил

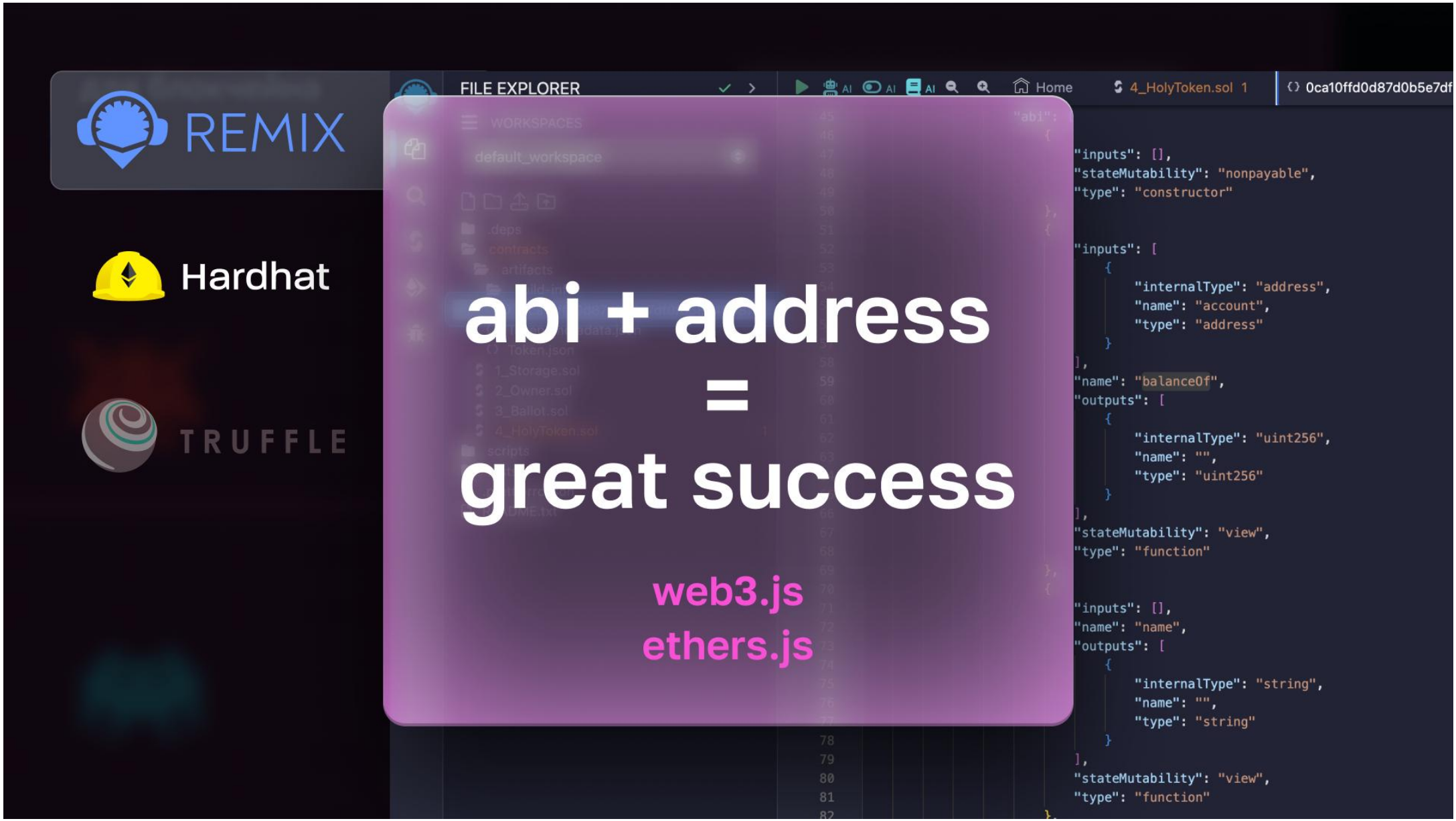

Получить
Account 1
0x9af60907001Ff1A93eD5d908CeDD8606945bd6d3
Скопировать в буфер обмена
Импорт токенов
Обновить список
Поддержка MetaMask
ГУГЛИТЕ SEPOLIA FAUCET
Account 1
0x9af60...bd6d3
0.5 SepoliaETH
Buy & Sell
Отправить
Своп
Мост
Portfolio
Недоступно в этой сети
Токены
SepoliaETH
SepoliaETH
0.5 SepoliaETH
Импорт токенов
Обновить список
Поддержка MetaMask


REMIX
Hardhat
TRUFFLE
FILE EXPLORER
WORKSPACES
default_workspace
.deps
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
4_HolyToken.sol
scripts
tests
.prettierrc.json
README.txt
Home
4_HolyToken.sol
pragma solidity ^0.8.0;
contract Token {
string public name = "Holy Token";
string public symbol = "HOLY";
uint256 public supply = 1000000;
address public owner;
mapping(address => uint256) balances;
constructor() { infinite gas 391600 gas
// The supply is assigned to the transaction sender, which is the acco
// that is deploying the contract.
balances[msg.sender] = supply;
owner = msg.sender;
}
function transfer(address to, uint256 amount) external { infinite gas
require(balances[msg.sender] >= amount, "Not enough tokens");
balances[msg.sender] -= amount;
balances[to] += amount;
}
function balanceOf(address account) external view returns (uint256) {
return balances[account];
}
}


REMIX
Hardhat
TRUFFLE
SOLIDITY COMPILER
COMPILER
0.8.0+commit.c7dfd78e
Include nightly builds
Auto compile
Hide warnings
Advanced Configurations
Compile 4_HolyToken.sol
Compile and Run script
CONTRACT
Token (4_HolyToken.sol)
Publish on Ipfs
Publish on Swarm
Compilation Details
ABI
Bytecode
Warning: SPDX license identifier not provided in source file. Before publishing, consider adding a comment containing "SPDX-License-Identifier: <SPDX-
Home
4_HolyToken.sol 1
pragma solidity ^0.8.0;
contract Token {
string public name = "Holy Token";
string public symbol = "HOLY";
uint256 public supply = 1000000;
address public owner;
mapping(address => uint256) balances;
constructor() { infinite gas 391600 gas
// The supply is assigned to the transaction sender, which is the acco
// that is deploying the contract.
balances[msg.sender] = supply;
owner = msg.sender;
}
function transfer(address to, uint256 amount) external { infinite gas
require(balances[msg.sender] >= amount, "Not enough tokens");
balances[msg.sender] -= amount;
balances[to] += amount;
}
function balanceOf(address account) external view returns (uint256) {
return balances[account];
}
}


REMIX
Hardhat
TRUFFLE
DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Testnet - Sepolia
Sepolia (11155111) network
ACCOUNT
0x9af...bd6d3 (0.4991081
GAS LIMIT
Estimated Gas
Custom
3000000
VALUE
0
Wei
CONTRACT
Token - contracts/4_HolyToken.sol
evm version: istanbul
Deploy
Publish to IPFS
At Address
Load contract from Address
Transactions recorded 1
Pinned Contracts (network: 11155111)
Home
4_HolyToken.sol 1
pragma solidity ^0.8.0;
contract Token {
string public name = "Holy Token";
string public symbol = "HOLY";
uint256 public supply = 1000000;
address public owner;
mapping(address => uint256) balances;
constructor() { infinite gas 391600 gas
// The supply is assigned to the transaction sender, which is the acco
// that is deploying the contract.
balances[msg.sender] = supply;
owner = msg.sender;
}
function transfer(address to, uint256 amount) external { infinite gas
require(balances[msg.sender] >= amount, "Not enough tokens");
balances[msg.sender] -= amount;
balances[to] += amount;
}
function balanceOf(address account) external view returns (uint256) {
return balances[account];
}
}

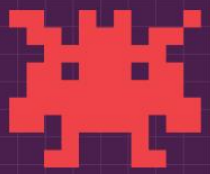

REMIX
Hardhat
TRUFFLE
FILE EXPLORER
WORKSPACES
default_workspace
.deps
contracts
artifacts
build-info
0ca10ffd0d87d0b5e7df0f058e46b874...
Token_metadata.json
Token.json
1_Storage.sol
2_Owner.sol
3_Ballot.sol
4_HolyToken.sol 1
scripts
tests
.prettierrc.json
README.txt
Home
4_HolyToken.sol 1
0ca10ffd0d87d0b5e7df
"abi": [
{
"inputs": [],
"stateMutability": "nonpayable",
"type": "constructor"
},
{
"inputs": [
{
"internalType": "address",
"name": "account",
"type": "address"
}
],
"name": "balanceOf",
"outputs": [
{
"internalType": "uint256",
"name": "",
"type": "uint256"
}
],
"stateMutability": "view",
"type": "function"
},
{
"inputs": [],
"name": "name",
"outputs": [
{
"internalType": "string",
"name": "",
"type": "string"
}
],
"stateMutability": "view",
"type": "function"


REMIX
Hardhat
TRUFFLE
abi + address
=
great success
web3.js
ethers.js
FILE EXPLORER
Home
4_HolyToken.sol 1
0ca10ffd0d87d0b5e7df
"inputs": [],
"stateMutability": "nonpayable",
"type": "constructor"
"inputs": [
{
"internalType": "address",
"name": "account",
"type": "address"
}
],
"name": "balanceOf",
"outputs": [
{
"internalType": "uint256",
"name": "",
"type": "uint256"
}
],
"stateMutability": "view",
"type": "function"
"inputs": [],
"name": "name",
"outputs": [
{
"internalType": "string",
"name": "",
"type": "string"
}
],
"stateMutability": "view",
"type": "function"

для блокчейна

Solidity

SDK



Bridge

Wallet

Swap crypto

Buy crypto

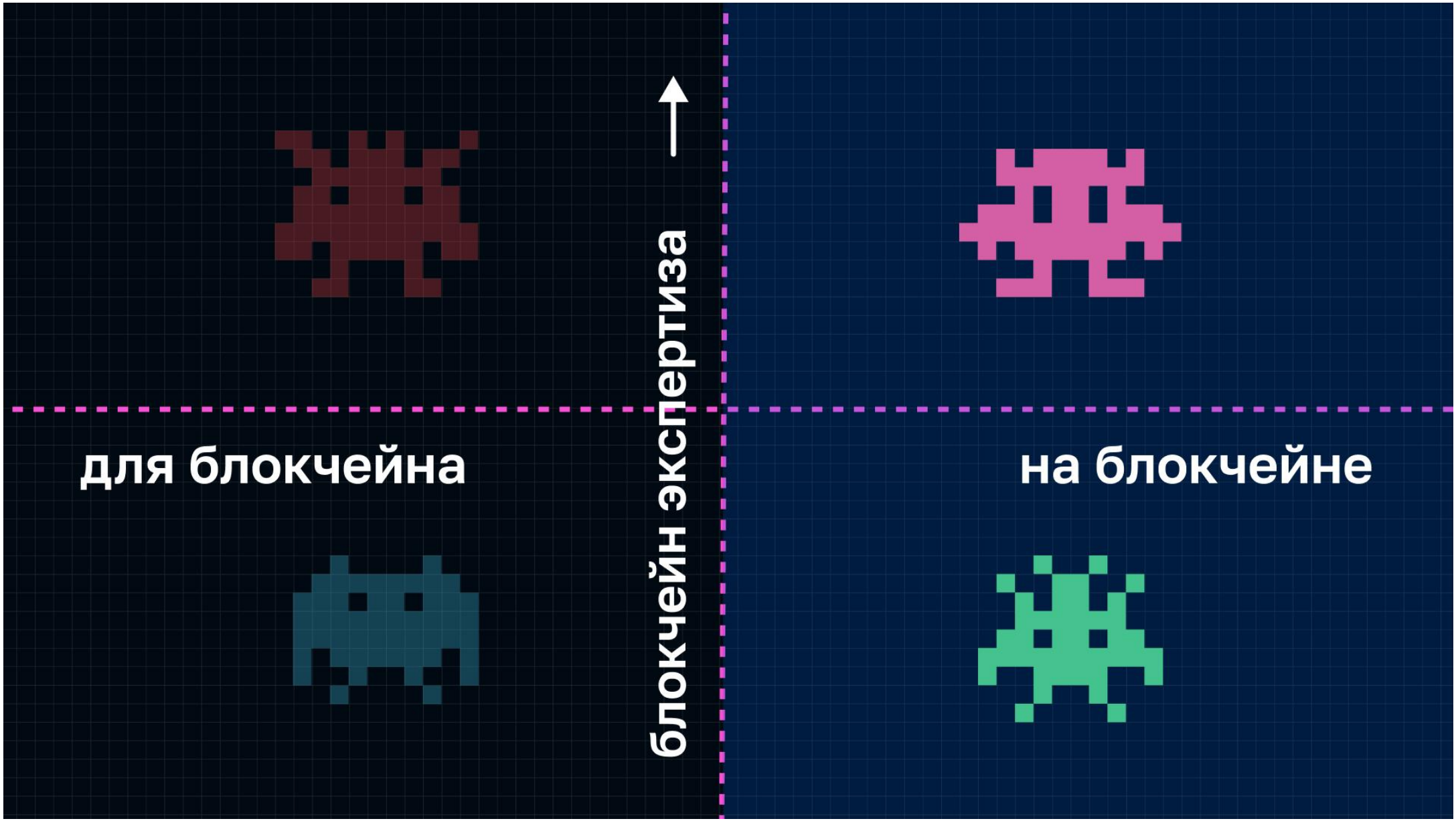
экспертиза

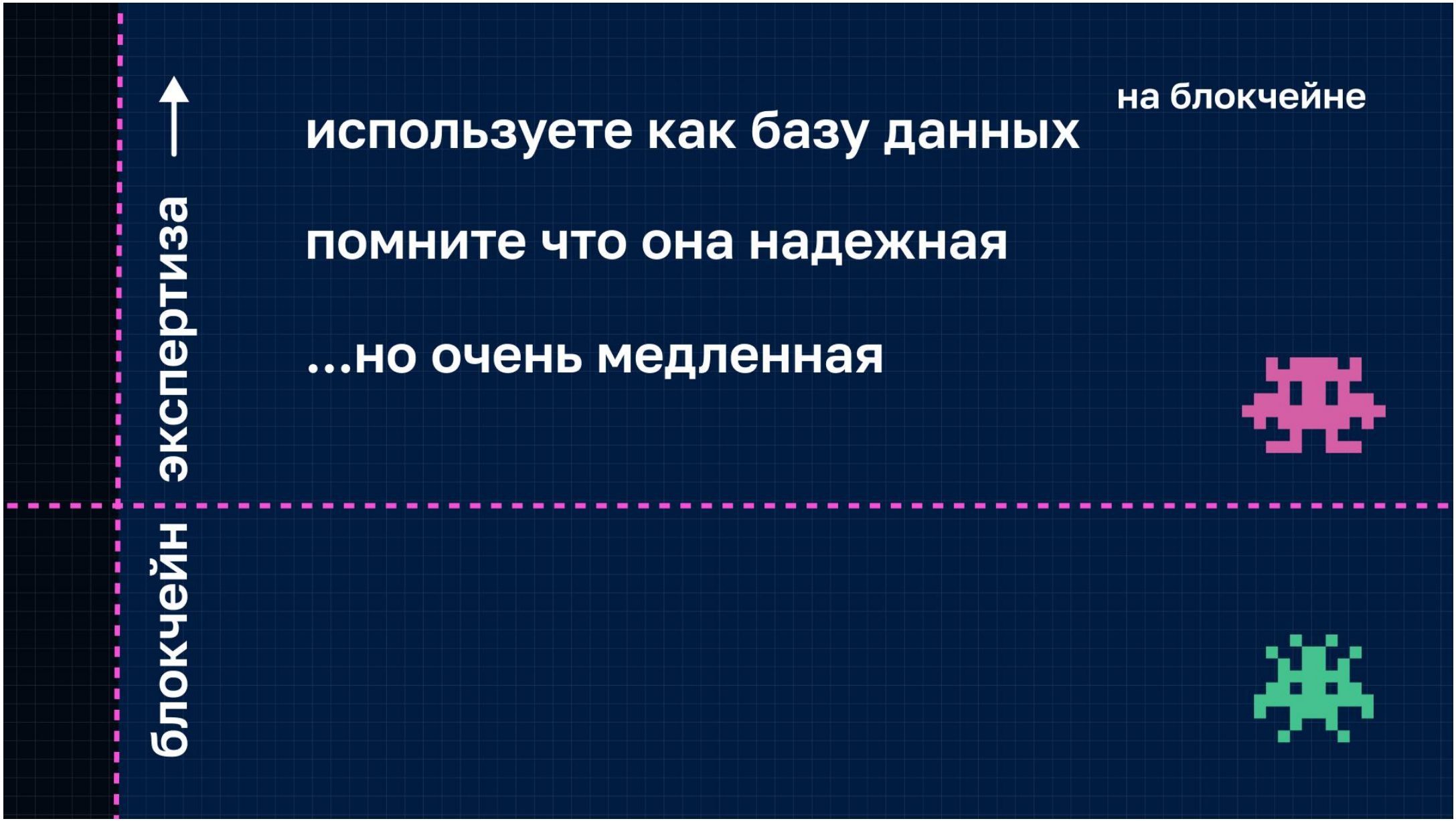
SDK

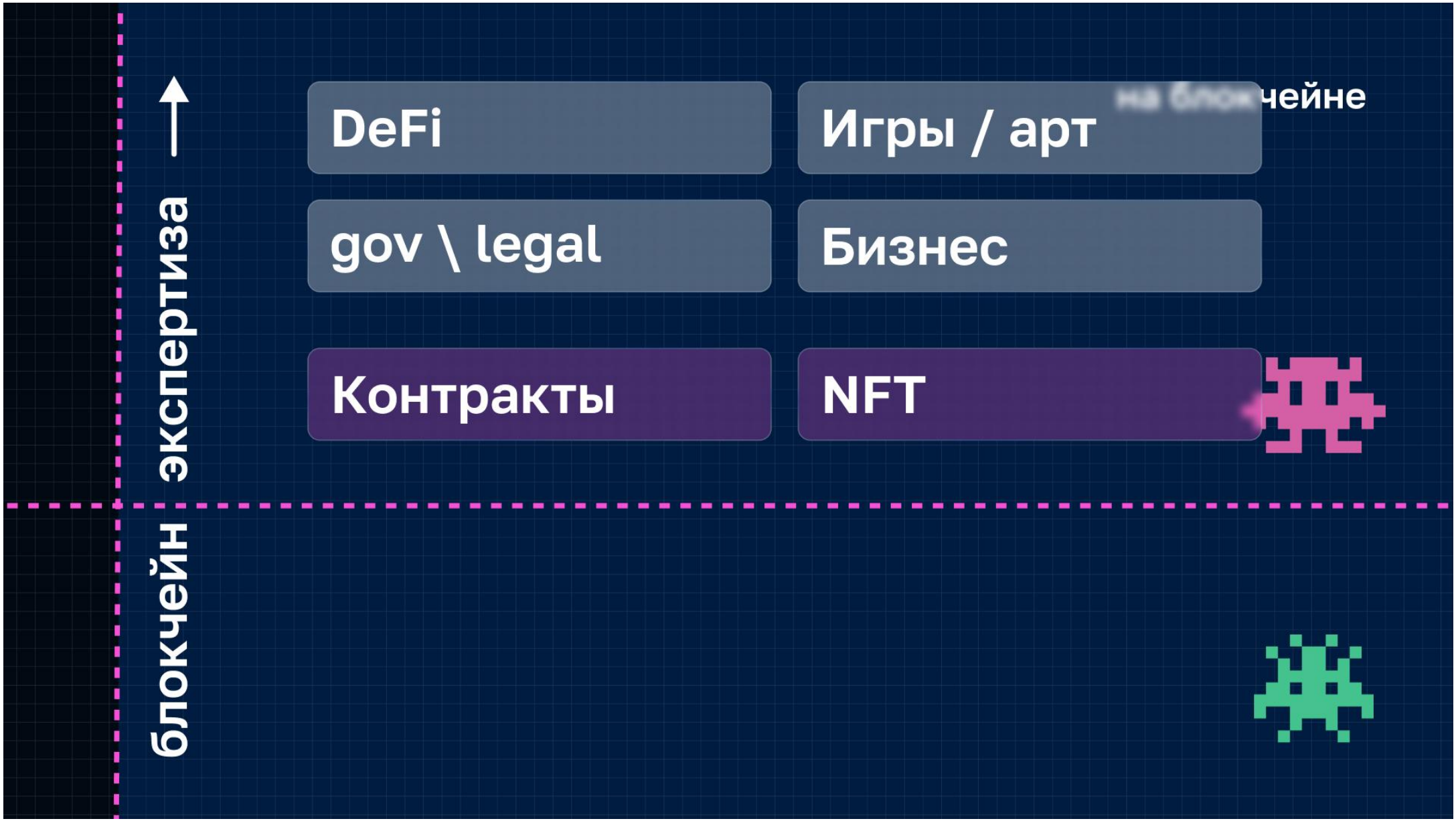
Staking, farming

Explorer/Scaner

блокчейн


блокчейн экспертиза
для блокчейна
на блокчейне


экспертиза
блокчейн
на блокчейне
используете как базу данных
помните что она надежная
...но очень медленная


экспертиза
блокчейн
DeFi
Игры / арт
чейне
gov \ legal
Бизнес
Контракты
NFT

cryptozombies.io

Solidity
by Example

solidity-by-example.org

на блокчейне

блокчейн экспертиза

Первый специальный случай

Разработка в блокчейне без блокчейна

Второй специальный случай

Мультичейн

Третий специальный случай

Сделай блокчейн сам

Советы и заветы

Мемы кончились

Советы

🚀 ищите идеи в простых механиках

🐣 делайте микросервисы, создавайте экосистемы

🧸 скорость > проработанность

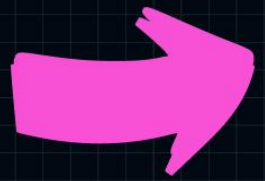
🧸 следите за грантами, хакатонами

Заветы

- 🐣 1,000,000 лучше чем 1,000,000 долларов
- 🧸 дружите с продактами и биздевами

Виталий
Панькин

6 отзывы
5 очень
4 надо
3 мне
2 бы
1 🙂



@vitpankin в телеге